

Driver-Automation Indirect Shared Control of Highly Automated Vehicles with Intention-Aware Authority Transition

Renjie Li, Yanan Li, Shengbo Eben Li, Etienne Burdet and Bo Cheng

Abstract—Shared control is an important approach to avoid the driver-out-of-the-loop problems brought by imperfect autonomous driving. Steer-by-wire technology allows the mechanical decoupling between the steering wheel and the road wheels. On steer-by-wire vehicles, the automation can join the control loop by correcting the driver steering input, which forms a new paradigm of shared control. The new framework, under which the driver indirectly controls the vehicle through the automation's input transformation, is called indirect shared control. This paper presents an indirect shared control system, which realizes the dynamic control authority allocation with respect to the driver's authority intention. The simulation results demonstrate the effectiveness and benefits of the proposed control authority adaptation method.

I. INTRODUCTION

Autonomous vehicles can entirely relieve human drivers of tedious manual operations and significantly cut down traffic accidents. In spite of these prospects, public concerns have arisen in view of the ensuing safety, ethics, and liability issues brought by autonomous driving [1]. In addition, previous studies have found that drivers being out of the loop is likely to cause human factor problems such as loss of situation awareness, over-reliance, distrust, *etc* [2], [3]. *Shared control* is a potential approach to resolve the predicament, which is defined as the vehicle control authority being continuously shared by, rather than unilaterally undertaken and rigidly switched between, the driver and the automation [4], [5].

The benefits of shared control come from multiple aspects. On one hand, shared control enables drivers to receive the assistance from the automation while remaining in the loop, thereby eliminating the latent out-of-the-loop risks. On the other hand, it incorporates the capabilities of human drivers into vehicle control, which largely expands the scope of situations that the automation can handle. One paradigm of shared control that has been extensively studied is the haptic shared control, wherein the driver and the automation collaboratively control the vehicle through haptic feedback and physical interaction [4]. Haptic shared control is generally implemented on traditional vehicles with mechanical steering systems, where the automation takes control by applying an

additional torque on the steering column. The application scenarios and benefits of haptic shared control are surveyed in [6]. *Steer-by-wire* technology allows the mechanical decoupling between the steering wheel and the road wheels. On steer-by-wire vehicles, shared control can be realized in an alternative form, in which the automation can join the control loop by transforming the driver input according to a predesignated algorithm. Under such a structure, the control flexibility of the automation is largely increased because the driver input can be arbitrarily modified and even canceled if the automation deems it necessary. In this sense, the driver can only control the vehicle indirectly through the automation's input transformation. In light of this property, such a shared control paradigm is called *indirect shared control*.

A central element of shared control is the control authority allocation. Previous studies with haptic shared control generally adopt a fixed authority allocation [7]–[9]. However, Mars *et al.* [10] argue that the automation's authority should depend on the traffic context. In addition, Abbink *et al.* [4] and Flemisch *et al.* [11] suggest that the control authority allocation should adapt to the driver's intention. For example, the driver may imply a higher control authority to secure the vehicle control if the automation's control ability is impaired, and a lower control authority if the situation can be well handled by the automation. Achieving efficient adaptable control authority allocation requires to estimate the driver's desired control authority (or authority intention) in real-time. However, estimating the authority intention in shared control is not a trivial task [12], [13], due to its vagueness and the lack of knowledge of how it shapes the driver's behavior.

In this paper, we propose a driver-automation indirect shared control system, of which the control authority allocation is adaptable with respect to the driver's authority intention, which is identified in real-time. First, we adopt a weighted summation method as the input transformation algorithm, which explicitly parametrizes the control authority as the weight assigned to the corresponding input. Then, we model the driver's control strategy under a desired authority using model predictive control (MPC), which assumes that the driver claims the desired authority by incorporating it into the internal model. Lastly, a sliding-window least-squares estimator is used to continuously identify the driver's desired authority, the output of which is filtered before being used for control authority adaptation.

The remnant of this paper is organized as follows. Section II proposes an indirect shared control system. Section III formulates the control strategies of the driver and the au-

This research is supported by NSF China with U1664263, NSF China with 51575293 and 51622504, and National Key R&D Program in China with 2016YFB0100906. All correspondence should be sent to S. Li (Email: lisb04@gmail.com).

R. Li, S. Li and B. Cheng are with the State Key Laboratory of Automotive Safety and Energy, Tsinghua University, Beijing, 100084, China (e-mail: lirenaxe@gmail.com)

R. Li, Y. Li and E. Burdet are with the Department of Bioengineering, Imperial College London, SW7 2AZ, UK

tomation under the proposed indirect shared control. Section IV presents an intention-aware method to dynamically adapt the control authority allocation. Section V illustrates the simulation results and Section VI is the conclusion.

II. INDIRECT SHARED CONTROL

The general framework of indirect shared control is shown in Fig. 1, where r_D and r_A are the reference paths of the driver and the automation, respectively, u_D is the driver's commanded steering input, and u is the actual steering input. u_D and u are connected by the transformation function $u = g(x, u_D, r_A)$. x is the vehicle state, and $\dot{x} = f(x, u)$ is the chosen vehicle model.

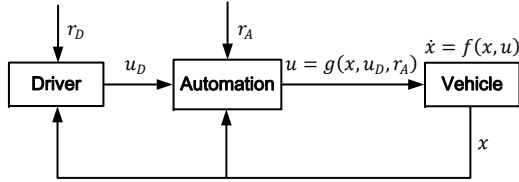


Fig. 1. Block diagram of driver-automation indirect shared control

A. Vehicle Model

In this paper, we use the linearized dynamic bicycle model to approximate vehicle dynamics, as shown in Fig. 2. The dynamic bicycle model has been broadly adopted in literature, which assumes that the tires work in a linear region [14], [15]. If the vehicle has a constant longitudinal velocity U , the model is expressed as a linear time-invariant equation

$$\begin{aligned} \dot{x}(t) &= A_c x(t) + B_c u(t) \\ z(t) &= C_c x(t) \end{aligned} \quad (1)$$

where the state $x(t) = [v(t) \ \omega(t) \ y(t) \ \psi(t)]^T$ (v —

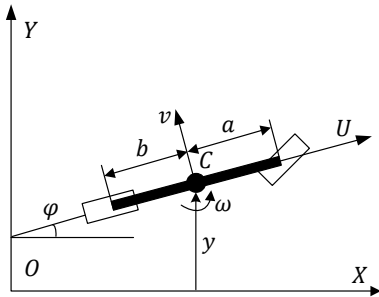


Fig. 2. Linearized dynamic bicycle model

lateral velocity, ω —yaw rate, y —lateral displacement, ψ —yaw angle), $u(t) = \delta(t)$ is the steering wheel angle, and $z(t) = [y(t) \ \psi(t)]^T$ is the output. A_c and B_c are constant

matrices:

$$\begin{aligned} A_c &= \begin{bmatrix} \frac{-(C_f + C_r)}{mU} & \frac{-(aC_f - bC_r)}{mU} - U & 0 & 0 \\ \frac{-(aC_f - bC_r)}{I_z U} & \frac{-(a^2 C_f + b^2 C_r)}{I_z U} & 0 & 0 \\ 1 & 0 & 0 & U \\ 0 & 1 & 0 & 0 \end{bmatrix} \\ B_c &= \begin{bmatrix} C_f & aC_f & 0 & 0 \\ i_s m & i_s I_z & 0 & 0 \end{bmatrix}^T, C_c = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \quad (2)$$

where C_f is the front cornering stiffness, C_r is the rear cornering stiffness, m is the vehicle mass, a is the distance from center of mass to front axle, b is the distance from center of mass to rear axle, I_z is the polar moment of inertia, and i_s is the steering ratio. The continuous-time vehicle model (1) can be converted to a discrete-time form by discretization, i.e.,

$$\begin{aligned} x(k+1) &= A x(k) + B u(k) \\ z(k) &= C x(k), \end{aligned} \quad (3)$$

where k denotes discrete time step. A , B and C are converted from A_c , B_c and C_c , respectively. The conversion is subject to a sampling time T .

B. Transformation Algorithm

The automation's input transformation algorithm $u = g(x, u_D, r_A)$ should ensure that both the interests of the driver and the automation are respected. We follow the weighted summation method used in [16], i.e.,

$$u(k) = \lambda_D u_D(k) + \lambda_A u_A(k), \quad \lambda_D, \lambda_A \geq 0, \quad (4)$$

where u_A is the automation's desired input, λ_D and λ_A are the weights assigned to the driver input and the automation's desired input, respectively. The advantage of this method is that the authorities of the driver and the automation can be explicitly parameterized as their corresponding weight. Basically, λ_D and λ_A are required to satisfy $\lambda_D + \lambda_A = 1$ to avoid conflict and overshooting. Thus, the shared control law (4) can be rewritten by denoting $\lambda_D \equiv \lambda$ and $\lambda_A \equiv 1 - \lambda$:

$$u(k) = \lambda u_D(k) + (1 - \lambda) u_A(k), \quad 0 \leq \lambda \leq 1. \quad (5)$$

III. STRATEGIES FORMULATION

In this section we first design the automation's control strategy to calculate u_A . Then, we model the driver's control strategy u_D under a desired control authority. This work lays foundation for the estimation of the driver's authority intention in Section IV.

A. Automation's Control Strategy

The automation adopts unconstrained MPC to calculate its desired input u_A at each step. MPC has been successfully

implemented in autonomous path tracking [17], [18]. The strategy is stated as

$$u_A(k) = \mathbf{u}_{A,(1)}^*(k) \quad (6a)$$

$$\mathbf{u}_A^*(k) = \arg \min_{\{u_A(k), u_A(k+1), \dots, u_A(k+N_c-1)\}} J_A(k) \quad (6b)$$

$$\text{s.t. } x(k+1) = Ax(k) + Bu_A(k) \quad (6c)$$

$$z(k) = Cx(k) \quad (6d)$$

where N_c is the control horizon, \mathbf{u}_A^* is the optimal input sequence over the control horizon, and $\mathbf{u}_{A,(1)}^*$ denotes the first element of \mathbf{u}_A^* . J_A is the automation's cost function designed as

$$J_A(k) \equiv \sum_{i=1}^{N_p} \|z(k+i) - r_A(k+i)\|_{Q_A}^2 + \sum_{i=0}^{N_u-1} \|u_A(k+i)\|_{R_A}^2 \quad (7)$$

where r_A is the automation's reference path at each step, N_p is the predictive horizon, and Q_A and R_A are constant weighting matrices of appropriate dimensions. This cost function penalizes path-tracking error and control input. u_A can be obtained by employing existing solutions for unconstrained MPC, as introduced in [19]:

$$u_A(k) = e_1^T \mathcal{K}_A \varepsilon_A(k), \quad (8)$$

where $e_1^T = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}$, $\mathcal{K}_A = \begin{bmatrix} \sqrt{Q_A} \Theta \\ \sqrt{R_A} \end{bmatrix}^\dagger \begin{bmatrix} \sqrt{Q_A} \\ 0 \end{bmatrix}$ (\dagger denotes pseudo-inverse) is a constant matrix, and the vector $\varepsilon_A(k) = \mathbf{r}_A(k) - \Phi x(k)$ is associated with the automation's reference path up to the predictive horizon and the current vehicle state. For the detailed expressions of $\mathbf{r}_A(k)$, Q_A , and R_A , see Appendix.

B. Driver's Control Strategy Under a Desired Authority

It has been validated that humans intend to construct an internal model of motor control tasks [20]. Therefore, some studies suggest that the driver steering behavior for path following can be modeled by MPC [15], [21], [22], where the driver's internal model of the vehicle dynamics serves as the predictor. In indirect shared control, the direct link between the driver and the vehicle motion is cut off by the automation. However, humans have an extraordinary ability to learn from and adapt to unknown systems. Therefore, we reasonably outline two basic assumptions:

Assumption 1: the driver can acquire the automation's input transformation (5) after motor learning and adaptation.

Assumption 2: the driver declares the authority intention by substituting the desired authority weight into (5), and uses it as the new predictor for path following.

Based on the above assumptions, the driver's control strategy can be stated as

$$u_D(k) = \mathbf{u}_{D,(1)}^*(k) \quad (9a)$$

$$\mathbf{u}_D^*(k) = \arg \min_{\{u_D(k), u_D(k+1), \dots, u_D(k+N_c-1)\}} J_D(k) \quad (9b)$$

$$\text{s.t. } u(k) = \lambda^* u_D(k) + (1 - \lambda^*) u_A(k) \quad (9c)$$

$$x(k+1) = Ax(k) + Bu(k) \quad (9d)$$

$$z(k) = Cx(k) \quad (9e)$$

where N_c is the control horizon which is assumed identical to the automation's, \mathbf{u}_D^* is the optimal input sequence up to the control horizon, and J_D is the driver's cost function. λ^* is the driver's desired authority weight. Note a special case $\lambda^* = \lambda$, which means that the driver and the automation agrees on the current control authorities and the indirect shared control reaches an equilibrium. For the driver's cost function J_D , we follow the one defined in [15] to model manual path following behavior, which takes a similar form to (7) and can be expressed as

$$J_D(k) \equiv \sum_{i=1}^{N_p} \|z(k+i) - r_D(k+i)\|_{Q_D}^2 + \sum_{i=0}^{N_u-1} \|u_D(k+i)\|_{R_D}^2 \quad (10)$$

where r_D is the driver's reference path, N_p is the predictive horizon which is assumed identical to the automation's, and Q_D and R_D are constant weighting matrices of appropriate dimensions. Similar to the automation, the driver also seeks to minimize the path-tracking error and the control effort. We hereby directly give the solution of u_D of problem (9)–(10). The driver's control strategy u_D is derived as

$$u_D(k) = h(x(k), \mathbf{r}_D(k), \mathbf{w}_A(k), \lambda^*) = e_1^T \mathcal{K}_D \varepsilon_D(k), \quad (11)$$

where

$$\mathcal{K}_D = \begin{bmatrix} \lambda^* \sqrt{Q_D} \tilde{\Theta} \\ \sqrt{R_D} \end{bmatrix}^\dagger \begin{bmatrix} \sqrt{Q_D} \\ 0 \end{bmatrix} \quad (12)$$

and

$$\varepsilon_D(k) = \mathbf{r}_D(k) - \tilde{\Phi} x(k) - (1 - \lambda^*) \tilde{\Theta} \mathbf{w}_A(k). \quad (13)$$

The definitions of $\mathbf{r}_D(k)$, $\mathbf{w}_A(k)$, Q_D , R_D , $\tilde{\Theta}$, and $\tilde{\Phi}$ are listed in Appendix. The derivation of $u_D(k)$ refers to the authors' previous work [23].

IV. ESTIMATION AND ADAPTATION OF DRIVER'S AUTHORITY INTENTION

This section has two main parts: the first one is to continuously estimate the driver's desired authority λ^* from the observed vehicle state x and driver input u_D ; and the second is to adapt the actual control authority λ based on the estimate.

A. Authority Intention Estimation

We assume that the reference path of the driver is identical to the automation (thus the automation can precisely estimate the driver's reference path at each step), and the measurement error of the vehicle state x is negligible. Now, if the driver input u_D is perfectly known to the automation and the model (11) has no model error, λ^* can be immediately recovered from (11) by nonlinear optimization. This premise is however unrealistic and we have to assume a certain level of uncertainty in the observed u_D . Let us denote the observed driver input as \tilde{u}_D , and assume that the measurement noise and model error can be jointly represented by Gaussian white noise:

$$\tilde{u}_D(k) = h(x(k), \mathbf{r}_D(k), \mathbf{w}_A(k), \lambda^*) + \xi(k), \quad (14)$$

where $\xi(k) \sim \mathcal{N}(0, \sigma^2)$ (σ is the *a priori* standard deviation) and $E[\xi(i)\xi^T(j)] = 0$ for $i \neq j$.

To select an appropriate method to estimate λ^* from (14), we have to consider that: 1) the measurement function $h(x(k), \mathbf{r}_D(k), \mathbf{w}_A(k), \lambda^*)$ is highly nonlinear according to (11)–(13); (2) λ^* is time-varying and the variation is quite unpredictable. For such nonlinear estimation problems, blockwise methods (such as nonlinear least-squares regression) generally outperform recursive methods (such as extended Kalman filter) at the cost of a higher computational load [24]. In this paper, we focus on the estimation accuracy and thus choose the sliding window least-squares regression to estimate λ^* . The estimator works by minimizing the error between \tilde{u}_D and expected u_D within a sliding window. The error is defined as

$$\epsilon(\lambda; k) = \sum_{j=k-H+1}^k \left(\tilde{u}_D(j) - h(x(j), \mathbf{r}_D(j), \mathbf{w}_A(j), \lambda) \right)^2, \quad (15)$$

where H is the constant window length. The least-squares estimate for λ^* at each step is hence stated as

$$\hat{\lambda}(k) = \arg \min_{\lambda} \epsilon(\lambda; k), \text{ s.t. } 0 \leq \lambda \leq 1. \quad (16)$$

Eq. (16) is a constrained nonlinear least-squares problem, effective solutions of which include the trust-region reflective algorithm [25].

B. Control Authority Adaptation

After the driver's authority intention is estimated, a straightforward approach is to simply adapt λ to the estimate $\hat{\lambda}$ at each step. However, least-squares estimation is highly sensitive to the input data. Thus, $\hat{\lambda}$ is likely to oscillate even if the true λ^* is steady. Moreover, if λ changes at each step and the sampling time is small, the indirect shared control system will possibly become unstable. Taking these factors into account, there are two considerations for the control authority adaptation: 1) the estimate $\hat{\lambda}$ should be smoothed before being used for authority adaptation; 2) the time interval of authority adaption should not be too small to avoid system instability.

To address consideration 1), we use a moving averaging filter with rounding to filter the estimate $\hat{\lambda}$, which works as below:

$$\hat{\lambda}_f(k) = \left[\frac{1}{H_f} \sum_{j=k-H_f+1}^k \hat{\lambda}(j) \right]_{0.1}, \quad (17)$$

where H_f denotes the window length of the averaging filter, and $[\cdot]_{0.1}$ means rounding the number towards nearest tenth. The selection of H_f will inevitably confront some trade-off. A large H_f can well reject the estimation error if λ^* is static, but the response time is longer if λ^* changes, and *vice versa*.

To address consideration 2), we use a zero-order holder to apply control authority adaptation. The zero-order holder simply enlarges the time interval of the authority adaptation:

$$\lambda(k) = \begin{cases} \hat{\lambda}_f(k) & k = iN_z, i = 1, 2, \dots \\ \lambda(k-1) & \text{otherwise.} \end{cases}, \quad (18)$$

where N_z is the sampling interval of the zero-order holder.

H_f and N_z can be properly tuned through off-line simulation. Their appropriate values depend on the scale of ξ and the performance preference of the designer. The control authority adaptation method can be summarized in Algorithm 1.

Algorithm 1: Control Authority Adaptation

Input: Observed driver input \tilde{u}_D , vehicle state x , driver reference path r_D , and w_A calculated from automation reference path r_A

Result: Adapt control authority λ

begin

 initialize $H, H_f, N_z, k \leftarrow \max(H, H_f)$

while $u_D(k), x(k), r_D(k), w_A(k)$ not terminated **do**

 define function $\epsilon(\lambda) = \sum_{j=k-H+1}^k \left(\tilde{u}_D(j) - h(x(j), \mathbf{r}_D(j), \mathbf{w}_A(j), \lambda) \right)^2$

$\hat{\lambda}(k) \leftarrow \arg \min_{\lambda} \epsilon(\lambda), \text{ s.t. } 0 \leq \lambda \leq 1$

if $k \in \{iN_z\}, i = 1, 2, \dots$ **then**

$\hat{\lambda}_f(k) \leftarrow \left[\frac{1}{H_f} \sum_{j=k-H_f+1}^k \hat{\lambda}(j) \right]_{0.1}$

$\lambda(k) \leftarrow \hat{\lambda}_f(k)$

else

$\lambda(k) \leftarrow \lambda(k-1)$

$k \leftarrow k + 1$

V. SIMULATION

In this section we present the simulation results to validate the proposed control authority adaptation method. In Section V-A, we will verify the driver authority intention estimator proposed in Section IV-A. We are particularly interested in the performance of the estimator with respect to the driver's different desired authorities. In Section V-B, we

TABLE I
SIMULATION PARAMETERS

Front wheel cornering stiffness	C_f	12000 [N/rad]
Rear wheel cornering stiffness	C_r	8000 [N/rad]
Distance from mass center to front axle	a	0.92 [m]
Distance from mass center to rear axle	b	1.38 [m]
Mass	m	1200 [kg]
Polar moment of inertia	I_z	1500 [kg·m ²]
Steering ratio	i_s	16
Longitudinal velocity	U	20 [m/s]
Sampling time	T	0.02 [s]
Predictive horizon	N	50
Automation weighting matrix	Q_A	$\begin{bmatrix} 1.5 & \\ & 0.6 \end{bmatrix}$
Driver weighting matrix	Q_D	$\begin{bmatrix} 0.16 & \\ & 0.06 \end{bmatrix}$

will investigate the effectiveness and benefits of the authority adaptation algorithm in Section IV-B. The simulations are performed on Matlab. The constrained nonlinear least-squares problem (16) is solved using the Matlab function *lsqnonlin*. The designed reference path for both the driver and the automation in the global coordinate system is shown in Fig. 3. If the longitudinal velocity U of the vehicle is constant, the reference path can be transformed to a temporal trajectory through the conversion $t = X/U$. The parameters for vehicle dynamics and MPC strategies are listed in Table I.

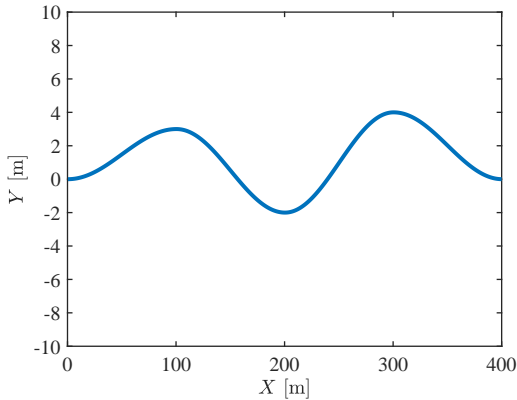


Fig. 3. Reference path

A. Authority Intention Estimation

We hereby examine the performance of the least-squares estimator. For the moment, we assume that the driver's desired authority λ^* is unchanged during each simulation and there is no adaptation of λ . The standard deviation of the measurement error of u_D is $\sigma = 0.002$ and the actual driver authority is $\lambda = 0.5$. The window length of the estimator is $H = 50$. The estimation result of the least-squares estimator with respect to the driver's desired

authority is shown in Fig. 4. We can see from Fig. 4 that the estimator performs well if λ^* is above 0.5. When λ^* is below 0.3, the estimation accuracy deteriorates as the result becomes oscillatory. The reason is that a lower λ^* generates a small u_D according to (11), and a small u_D makes \tilde{u}_D inaccurate due to the measurement noise. This result supports the analysis in Section IV-B that the estimate $\hat{\lambda}$ should be smoothed before being used for authority adaptation.

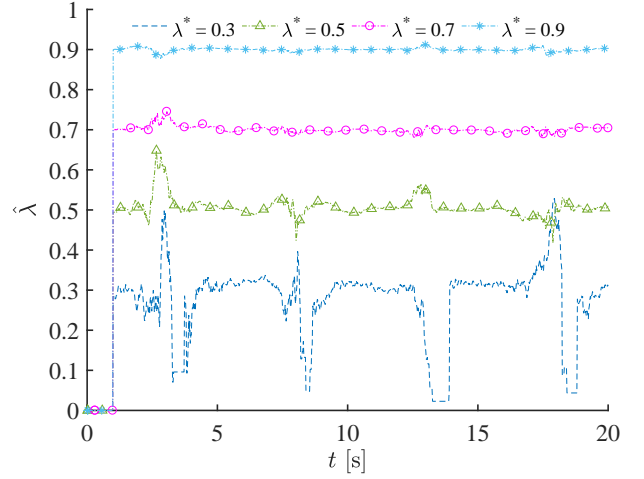


Fig. 4. Estimator performance with different driver desired authorities

B. Control Authority Adaptation

Now, we verify the effectiveness of the control authority adaptation method proposed in Section IV-B. In the meantime, we also investigate its benefits compared with static authority. The simulation is performed in two contrary scenarios. The window length of the moving averaging filter is $H_f = 2H = 100$. The sampling interval of the zero-order holder is $N_z = H = 50$.

1) *Driver Increases the Desired Authority*: In this scenario, we assume that the path-tracking ability of the automation is somehow comprised. This may be due to an impaired environment perception or to terrible road conditions. After knowing this, the driver requests an authority lift (from 0.2 to 0.9) to maintain the vehicle path-following performance. The impairment of the path-tracking ability of the automation is modeled by setting $Q_A = \text{diag}(0.015, 0.016)$. The driver's desired authority λ^* , least-squares estimate $\hat{\lambda}$, and the actual λ with and without adaptation (static authority) are shown in Fig. 5(a). Fig. 5(a) illustrates that λ well matches the driver's desired authority λ^* at steady state. When λ^* is stepped from 0.2 to 0.9, λ converges to the new λ^* in 3 seconds. The associated vehicle lateral displacement is illustrated in Fig. 5(b). From Fig. 5(b) we can see that if the authority is static, even if the driver intentionally lifts the desired authority, he cannot improve the path-tracking performance. However, this problem can be solved by adaptable authority weight. From Fig. 5(b) we can also see that although λ adaptation is discontinuous due to the filtering, the vehicle motion is not significantly affected as the trajectory remains smooth.

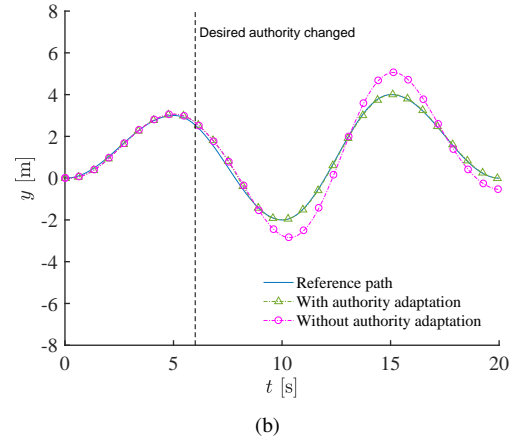
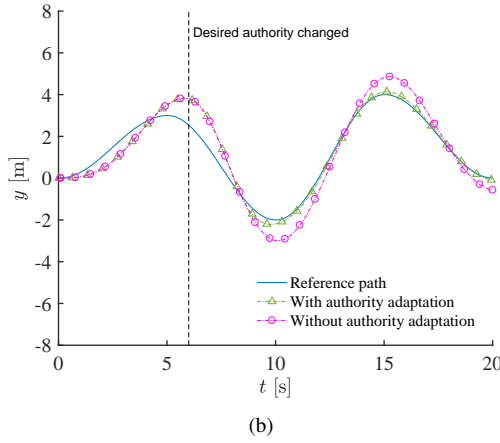
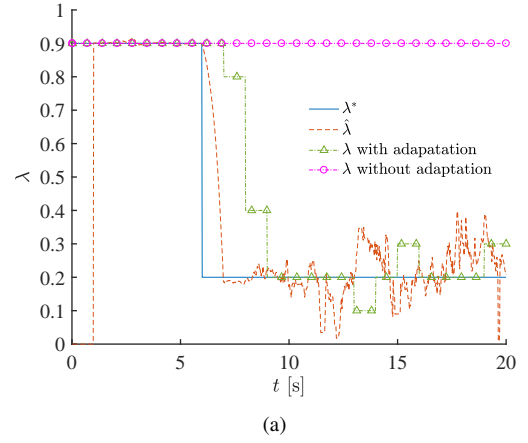
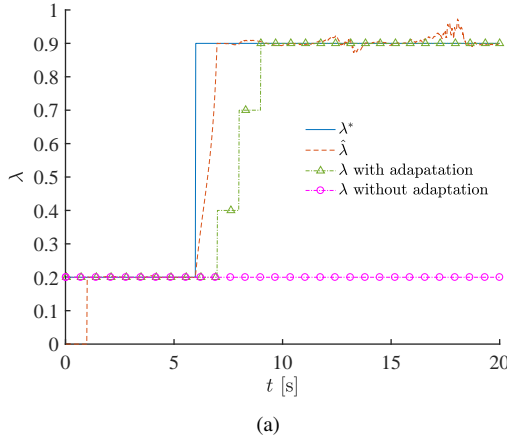


Fig. 5. Driver increases the desired authority: a) authority profile; b) vehicle lateral displacement with/without authority adaptation.

Fig. 6. Driver decreases the desired authority: a) authority profile; b) vehicle lateral displacement with/without authority adaptation.

2) *Driver Decreases the Desired Authority*: In this scenario, the driver initially takes almost full control of the vehicle ($\lambda^* = 0.9$), but then decides to release some control authority for relaxing ($\lambda^* = 0.2$). The driver's desired authority λ^* , least-squares estimate $\hat{\lambda}$, and the actual λ with adaptation and without adaptation (static authority) are depicted in Fig. 6(a). Fig. 6(a) also shows a satisfactory steady-state performance of the adaptation of λ , despite a slight oscillation (error = 0.1) when λ^* drops to 0.2. The associated vehicle lateral displacement is illustrated in Fig. 6(b), from which we can see that if the authority is static, the automation cannot automatically take over more control if the driver releases some control authority. This certainly affects the vehicle path-tracking performance. Once again, such a problem is avoided with authority adaptation

VI. CONCLUSION AND FUTURE WORKS

In this paper we have proposed a driver-automation indirect shared control system. The control authority allocation of the proposed system can be dynamically adapted with respect to the driver's authority intention. Simulation results have demonstrated the effectiveness and benefits of adaptable control authority compared with the conventional static one: the former guarantees a satisfying task performance by

combining complementary advantages of driver and automation in different situations. Future works include experiment validation (considering the uncertainties in modeling human driver's behaviors) and algorithm improvement to reduce computational load for practical implementation.

REFERENCES

- [1] J.-F. Bonnefon, A. Shariff, and I. Rahwan, "The social dilemma of autonomous vehicles," *Science*, vol. 352, no. 6293, pp. 1573–1576, 2016.
- [2] J. C. De Winter, R. Happee, M. H. Martens, and N. A. Stanton, "Effects of adaptive cruise control and highly automated driving on workload and situation awareness: A review of the empirical evidence," *Transportation research part F: traffic psychology and behaviour*, vol. 27, pp. 196–217, 2014.
- [3] N. Merat, A. H. Jamson, F. C. Lai, and O. Carsten, "Highly automated driving, secondary task performance, and driver state," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 54, no. 5, pp. 762–771, 2012.
- [4] D. A. Abbink, M. Mulder, and E. R. Boer, "Haptic shared control: smoothly shifting control authority?," *Cognition, Technology & Work*, vol. 14, no. 1, pp. 19–28, 2012.
- [5] L. Saleh, P. Chevrel, F. Claveau, J.-F. Lafay, and F. Mars, "Shared steering control between a driver and an automation: Stability in the presence of driver behavior uncertainty," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 974–983, 2013.
- [6] S. M. Petermeijer, D. A. Abbink, M. Mulder, and J. C. de Winter, "The effect of haptic support systems on driver performance: A literature survey," *IEEE transactions on haptics*, vol. 8, no. 4, pp. 467–479, 2015.

- [7] T. Brandt, T. Sattel, and M. BOhm, "Combining haptic human-machine interaction with predictive path planning for lane-keeping and collision avoidance systems," in *2007 IEEE Intelligent Vehicles Symposium*, pp. 582–587, IEEE, 2007.
- [8] M. Mulder, D. A. Abbink, and E. R. Boer, "Sharing control with haptics seamless driver support from manual to automatic control," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 54, no. 5, pp. 786–798, 2012.
- [9] B. Soualmi, C. Sentouh, J. Popieul, and S. Debernard, "Automation-driver cooperative driving in presence of undetected obstacles," *Control engineering practice*, vol. 24, pp. 106–119, 2014.
- [10] F. Mars, M. Deroo, and J.-M. Hoc, "Analysis of human-machine cooperation when driving with different degrees of haptic shared control," *IEEE transactions on haptics*, vol. 7, no. 3, pp. 324–333, 2014.
- [11] F. Flemisch, M. Heesen, T. Hesse, J. Kelsch, A. Schieben, and J. Beller, "Towards a dynamic balance between humans and automation: authority, ability, responsibility and control in shared and cooperative control situations," *Cognition, Technology & Work*, vol. 14, no. 1, pp. 3–18, 2012.
- [12] M. S. Erden and T. Tomiyama, "Human-intent detection and physically interactive control of a robot without force sensors," *IEEE Transactions on Robotics*, vol. 26, no. 2, pp. 370–382, 2010.
- [13] Y. Li and S. S. Ge, "Human-robot collaboration based on motion intention estimation," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 3, pp. 1007–1014, 2014.
- [14] C. C. MacAdam, "An optimal preview control for linear systems," 1980.
- [15] D. Cole, A. Pick, and A. Odhams, "Predictive and linear quadratic methods for potential application to modelling driver steering control," *Vehicle System Dynamics*, vol. 44, no. 3, pp. 259–284, 2006.
- [16] O. Manabu, T. Fujioka, N. Hashimoto, and H. Shimizu, "The application of rtk-gps and steer-by-wire technology to the automatic driving of vehicles and an evaluation of driver behavior," *IATSS research*, vol. 30, no. 2, pp. 29–38, 2006.
- [17] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *IEEE Transactions on control systems technology*, vol. 15, no. 3, pp. 566–580, 2007.
- [18] G. V. Raffo, G. K. Gomes, J. E. Normey-Rico, C. R. Kelber, and L. B. Becker, "A predictive controller for autonomous vehicle path tracking," *IEEE transactions on intelligent transportation systems*, vol. 10, no. 1, pp. 92–102, 2009.
- [19] J. M. Maciejowski, *Predictive control: with constraints*. Pearson education, 2002.
- [20] D. M. Wolpert, Z. Ghahramani, and M. I. Jordan, "An internal model for sensorimotor integration," *Science*, vol. 269, no. 5232, p. 1880, 1995.
- [21] S. D. Keen and D. J. Cole, "Application of time-variant predictive control to modelling driver steering skill," *Vehicle System Dynamics*, vol. 49, no. 4, pp. 527–559, 2011.
- [22] T. Qu, H. Chen, D. Cao, H. Guo, and B. Gao, "Switching-based stochastic model predictive control approach for modeling driver steering skill," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 365–375, 2015.
- [23] R. Li, Y. Li, S. Li, E. Burdet, and B. Cheng, "Indirect shared control of highly automated vehicles for cooperative driving between driver and automation." arXiv preprint <https://arxiv.org/abs/1704.00866>, 2017.
- [24] J. Jiang and Y. Zhang, "A novel variable-length sliding window blockwise least-squares algorithm for on-line estimation of time-varying parameters," *International journal of adaptive control and signal processing*, vol. 18, no. 6, pp. 505–521, 2004.
- [25] J. J. Moré and D. C. Sorensen, "Computing a trust region step," *SIAM Journal on Scientific and Statistical Computing*, vol. 4, no. 3, pp. 553–572, 1983.

APPENDIX NOTATIONS

$$\mathbf{r}_A(k) = \begin{bmatrix} r_A(k+1) \\ r_A(k+2) \\ \vdots \\ r_A(k+N) \end{bmatrix}, \quad \mathbf{r}_D(k) = \begin{bmatrix} r_D(k+1) \\ r_D(k+2) \\ \vdots \\ r_D(k+N) \end{bmatrix}$$

$$\mathcal{A} = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}, \quad \mathcal{B} = \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix}$$

$$\mathcal{C} = \begin{bmatrix} C \\ C \\ \ddots \\ C \end{bmatrix} \Bigg\}^N, \quad \Phi = \mathcal{C}\mathcal{A} = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^{N-1} \end{bmatrix}$$

$$\Theta = \mathcal{C}\mathcal{B} = \begin{bmatrix} CB & 0 & \cdots & 0 \\ CAB & CB & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{N-1}B & CA^{N-2}B & \cdots & CB \end{bmatrix}$$

$$\mathcal{Q}_A(\mathcal{Q}_D) = \begin{bmatrix} Q_A(Q_D) & & & \\ & Q_A(Q_D) & & \\ & & \ddots & \\ & & & Q_A(Q_D) \end{bmatrix} \Bigg\}^N$$

$$\mathcal{R}_A(\mathcal{R}_D) = \begin{bmatrix} R_A(R_D) & & & \\ & R_A(R_D) & & \\ & & \ddots & \\ & & & R_A(R_D) \end{bmatrix} \Bigg\}^N$$

$$\tilde{A} = A - (1 - \lambda^*)Be_1^T \mathcal{K}_A \Phi, \quad w_A(k) = e_1^T \mathcal{K}_A \mathbf{r}_A(k)$$

$$\tilde{A} = \begin{bmatrix} \tilde{A} \\ \tilde{A}^2 \\ \vdots \\ \tilde{A}^N \end{bmatrix}, \quad \tilde{B} = \begin{bmatrix} B & 0 & \cdots & 0 \\ \tilde{A}B & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{A}^{N-1}B & \tilde{A}^{N-2}B & \cdots & B \end{bmatrix}$$

$$\mathbf{w}_A(k) = \begin{bmatrix} w_A(k+1) \\ w_A(k+2) \\ \vdots \\ w_A(k+N) \end{bmatrix}, \quad \tilde{\Phi} = \mathcal{C}\tilde{A} = \begin{bmatrix} C\tilde{A} \\ C\tilde{A}^2 \\ \vdots \\ C\tilde{A}^N \end{bmatrix}$$

$$\tilde{\Theta} = \mathcal{C}\tilde{B} = \begin{bmatrix} CB & 0 & \cdots & 0 \\ C\tilde{A}B & CB & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C\tilde{A}^{N-1}B & C\tilde{A}^{N-2}B & \cdots & CB \end{bmatrix}.$$